Comparative study of Super Learner and Deep Learning as a Binary Classifier for Plastic Explosives Detection

Group Members: Jennifer Nasirumbi, Udayveer Singh Andotra, Andrew Walker

Course: 960:588 Data Mining, Fall 2024

Instructor: Professor Javier Cabrera

Summary

It was found that the SuperLearner ensemble and optimized Super Learner performed marginally better than the Deep Learning model for this binary classification problem of accurately predicting plastic explosives as per the data provided. All three methods had accuracy rates higher than 95%, when using no more than 20% of the training data to evaluate the model. Final predictions are made for the test dataset provided based on the Super Learner ensemble model optimized using Genetic Algorithm. All code files, datasets used and the final predicted dataset is attached in the appendix.

1. Introduction

The goal of the project was to compare two classification methods: **Super Learner** (an ensemble method) and **Deep Learning** (a neural network approach) in the development of a classifier that would provide approximately 0% miscalculation rate (No errors) in the prediction/detection of the presence of plastic explosives in suitcases using X-ray absorption spectrum data. Both methods were applied to the training data, evaluated using a validation set, and their performance compared based on misclassification rates and interpretability. The task is to detect which sets of data represent a bomb's presence in an airline suitcase. This is a binary classification problem with potentially exigent consequences, hence the need to be as accurate as possible. Presented in this report are two techniques used to assess and classify correctly what response variable values represent a bomb in a suitcase.

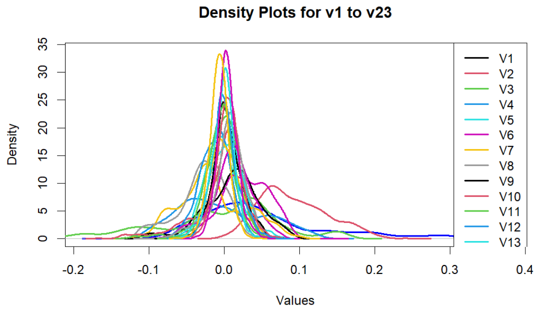
---

2. Data Overview

The dataset contains 23 variables representing the X-ray components of the spectrum, and the last variable indicates whether explosives are present (1) or absent (0). The objective is to build effective classifiers, compare their performances, and evaluate their ability to correctly identify suitcases with explosives. The dataset provided is pex23train.RDS, which includes 23 continuous features representing the X-ray absorption spectrum and a binary response variable (1 for explosive and 0 for non-explosive).

- **Features (X)**: The first 23 variables, which are discrete x-components of the X-ray spectrum.
- **Response (y)**: A binary variable indicating whether explosives are present (0) or absent (1).

The training data is split into **80% training** and **20% validation** subsets. The data variables show a distribution which is not deemed bad for modelling. There are no missing values but there are a few outliers.



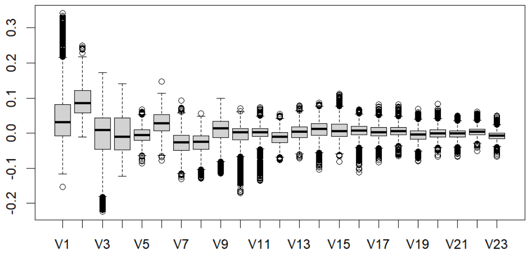


Figure0: Variable Distribution

---

3. Super Learner Results

For the SuperLearner ensemble method a combination of 13 popular classification methods/ base learners and the benchmark Discrete learner, SL-Mean were used to create a combined meta-model.

The following are the base learners that were included:

- **Linear Models:** GLM, glmnet
- **Ensemble Models:** Random Forest, XGBoost
- **Support Vector Machine:** SVM
- **Non-Linear Models:** KNN, Neural Network, Decision Trees (rpart), MARS, PPR
- **Discriminant Analysis:** LDA, QDA
- **Benchmark Model:** Mean of Y

The meta-SuperLearner was then evaluated against the individual Discrete Learners to assess model fit accuracy. The table1 and Figure1 below show the coefficients of base learners and the risk value associated with it as per the cross validated super learner. We are more inclined to go for low risk base learners. Analysis of the result of the superLearner shows that -

I. **Final SuperLearner Ensemble is the meta-learner that optimizes the objective function and includes the following Discrete Learners:**

  - XGBoost (SL.xgboost_All), K-Nearest Neighbors (SL.knn_All), Neural Network (SL.nnet_All), Recursive Partitioning Trees (Decision Trees) (SL.rpart_All), Projection Pursuit Regression (SL.earth_All), Quadratic Discriminant Analysis (SL.qda_All) were included in the final SuperLearner ensemble.
  - These models were included because they had the best combination of lowest risk i.e. least prediction errors and the highest coefficients i.e. they contributed accordingly to the SuperLearner ensemble model.

II. **Models with zero coefficient were excluded from the SuperLearner ensemble:**

  - SL.glm_All, SL.glmnet_All, SL.randomForest_All, SL.svm_All, SL.lda_All, SL.mean_All, SL.gam_All

Table1: 10-fold CV weight SuperLearner

Discrete Learner	Risk	Coef
SL.glm_All	0.009453414	0.00000000
SL.glmnet_All	0.003780415	0.00000000
SL.randomForest_All	0.006620838	0.00000000
SL.xgboost_All	0.005503828	0.05981727
SL.svm_All	0.005008650	0.00000000
SL.mean_All	0.231074215	0.00000000
SL.knn_All	0.004430063	0.02354782
SL.lda_All	0.015607159	0.00000000
SL.qda_All	0.001543190	0.80062409
SL.nnet_All	0.008564101	0.01948979
SL.rpart_All	0.013986243	0.04177866
SL.earth_All	0.009185756	0.05474236
SL.gam_All	0.009712644	0.00000000
SL.ppr_All	0.187379939	0.00000000

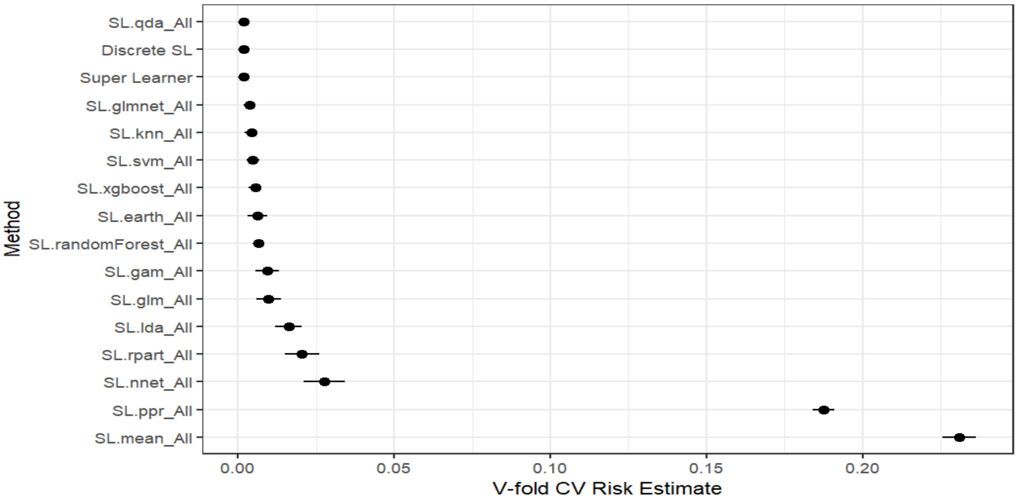
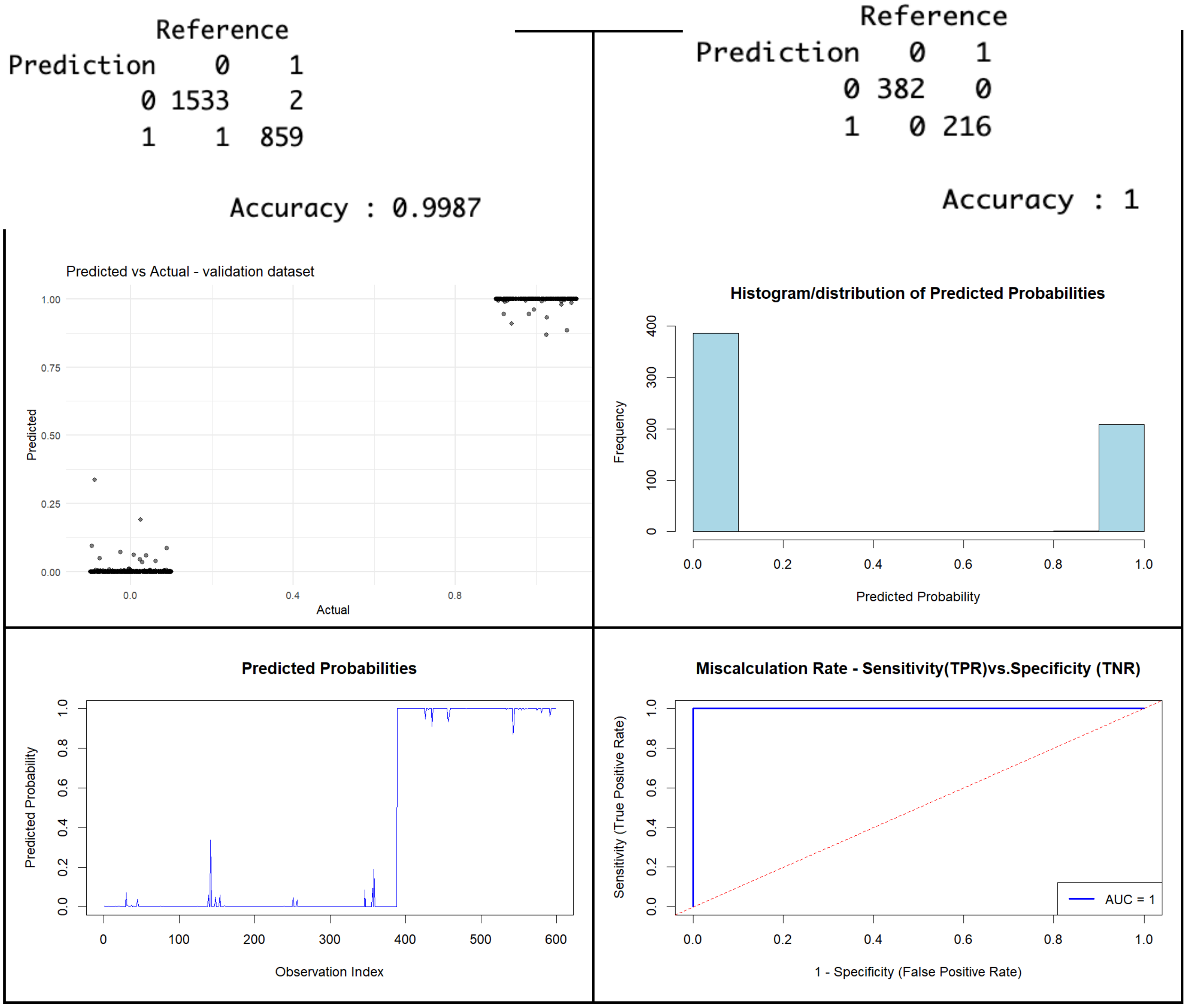


Figure1: Learners with predictive errors - using cross-validation to estimate the risk on future data

III.Prediction on the validation set from the Final SuperLearner Ensemble shows

Confusion matrix from the base SuperLearner function shows 100% accuracy for validation and the 99.87% for training (2 False positive and 1 False negative)-



```
          Reference
Prediction    0    1
         0 1533    2
         1    1  859

          Accuracy : 0.9987
```

```
          Reference
Prediction   0   1
         0 382   0
         1   0 216

          Accuracy : 1
```

Figure2: Super Learner Performance- Predicted vs actual probabilities and ROC curve

IV. Analysis of inclusion of weights in the base SuperLearner function and the effect of use of hyperparameters on the meta model generated from the ensemble model:

The performance of the SuperLearner ensemble was evaluated using **nested cross-validation** using the standard errors on the performance of the individual algorithms and that compared with the SuperLearner. The “Discrete SL” performed better than “Super Learner”, as shown in Figure1. The “Discrete SL” chooses the best single learner - SL.qda.

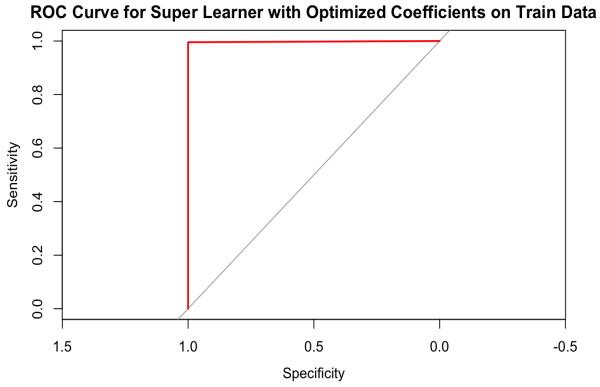
4. Super Learner optimised by Genetic Algorithm

Three Base Learners were selected based on the result of 10-fold cross validated Super Learner - GLM, SVM and XgBoost. GLM and SVM had coefficients zero while XgBoost showed better performance. The idea is to use Genetic Algorithm to optimize the coefficients of these base learners by using weights. Genetic Algorithms try to find the global maximum from population by using mutations and other control parameters to maximise the prediction accuracy using a fit function. The output from the fit function is a set of optimised weights which are then used to scale the coefficients of the base/un-optimised Super Learner.

We used a population size of 100, mutation rate 0.3, elitism as 2 and probability of crossover as 0.5. The global maximum for accuracy was achieved in 10 generations. Following are performance metrics - misclassification rate and ROC, for validation data.

Table3: Misclassification Rate:

	Base SuperLearner	**Weight Optimized SuperLearne**
Training	0.2088%	0.1670%
Validation	0.1672 %	0%



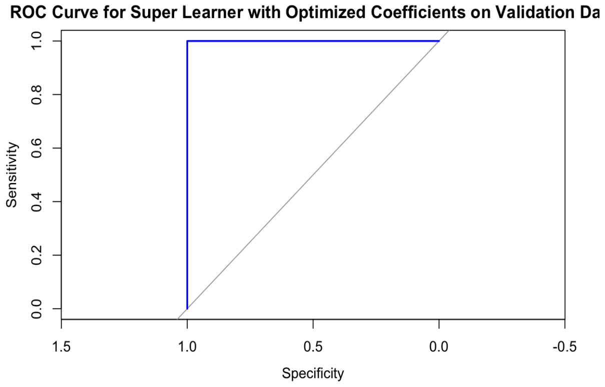
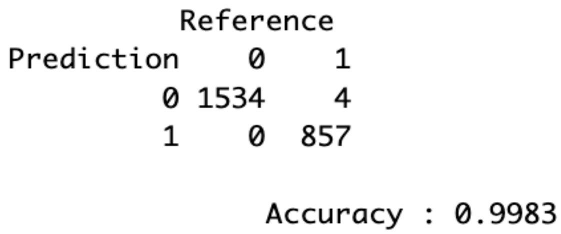


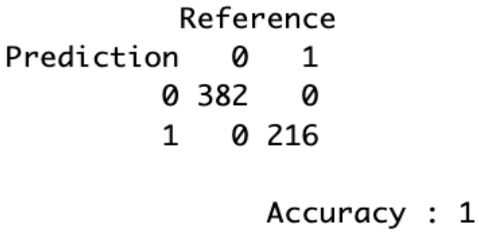
Figure3: GA optimized Super Learner ROC curve

Training: **Validation:**



```
          Reference
Prediction    0    1
         0 1534    4
         1    0  857

              Accuracy : 0.9983
```



```
          Reference
Prediction   0   1
         0 382   0
         1   0 216

              Accuracy : 1
```

The model predicted accurately for validation but was 99.83% accurate for training with 4 False positives.

5. Deep Learning

Deep learning utilized Multilayer Perceptron (MLP) with three layers including a sigmoidal output function.

Compared to other algorithms ANNs don’t require expert input during the feature design and engineering phase. They can learn the characteristics of the data and learn how to represent the data with features they extract on their own.
We used Sigmoid instead of RELU as ReLU excels in efficiency, gradient propagation, and scalability, sigmoid offers interpretability and suitability for certain tasks like binary classification.

The model was created, compiled and trained. The results and analysis will demonstrate its suitability for this project. Here also up to 20% of the training data was used for evaluation. Sequential model implemented where layers are added one after another in a linear stack, effective for classification problems.

Top layer has 64 neurons fully connected. ReLU (Rectified Linear Unit) activation function introduces non-linearity ensuring the model is capable of learning complex patterns. We added a dropout rate of 0.4, randomly sets 40% of the neurons to zero during training.

Second dense layer utilizes 32 neurons and ReLU activation. Added another dropout with a rate of 20%. Added two output classes (0 = explosive, 1= no explosive), and sigmoid activation function.

The input data for training, consisting of the feature matrix, utilized all 23 features. The response variable for training is in the categorical format necessary for binary classification.

Epochs was set to 50 (number of times the training data is passed through the model during training). Setting a higher number of epochs allows the model to learn more patterns in the data but increases the risk of overfitting if the model is trained too long. Following are the performance metrics on the validation data-

Table4: Deep Learning performance statistics

Sensitivity	1.0
Specificity	1.0
Pos Pred Value	1.0
Neg Pred Value	1.0
Prevalence	0.6488
Detection Rate	0.6488

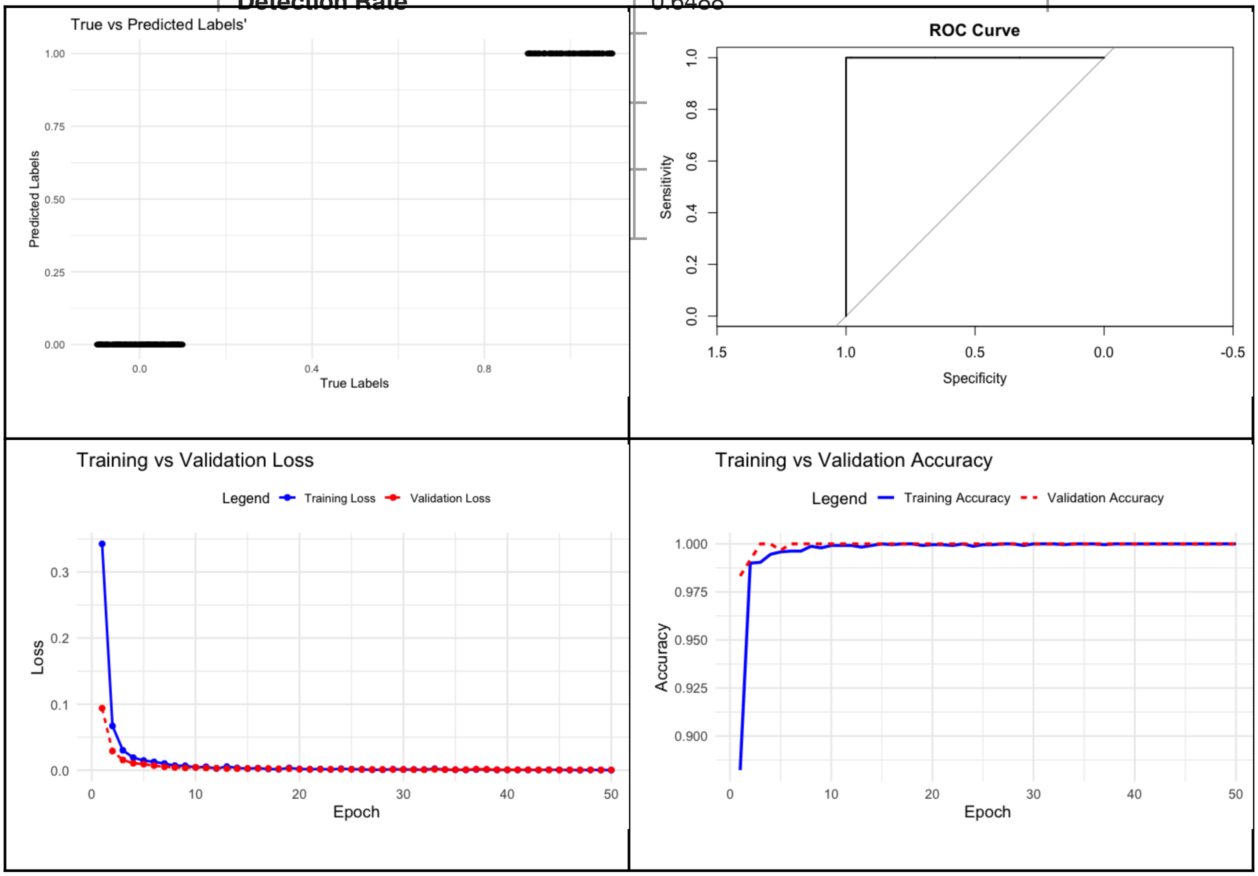


Figure4: Deep Learning Performance- Predicted vs actual labels, ROC curve and Loss/Accuracy for epochs

6. Comparison of the 3 methods was based on several performance metrics:

- **Accuracy**: The proportion of correctly classified instances.
- **Precision**: The proportion of true positive predictions out of all positive predictions.
- **Recall**: The proportion of true positives out of all actual positives.

6.1 Misclassification Rates

The **misclassification rate** is a simple measure of the proportion of incorrect predictions made by the classifier. Based on our results:

- **Super Learner Misclassification Rate**: 0%
- **Deep Learning Misclassification Rate**: 0.5%

This indicates that the Super Learner model performed slightly better in terms of overall classification accuracy.

6.2 Model Interpretability

- **Super Learner**: The Super Learner model is more interpretable than deep learning because it uses base learners like Logistic Regression, which provides straightforward interpretability in terms of coefficients. MARS and PPR can provide some insight into nonlinear relationships between features and the response.
- **Deep Learning**: The deep learning model, especially with a neural network, is less interpretable. While deep learning models may capture complex relationships in the data, they are often considered "black-box" models. It is difficult to understand how the model reaches its decisions, making it harder to interpret the decision boundaries.

---

7. Predictions on the Test Set

Using the trained models, we made predictions on the test set (pex23.test) and generated the following RDS file for evaluation:

- **File**: test_predictions.rds

The test set was used to further evaluate how well each model generalizes to new, unseen data. The predictions were saved for future evaluation.

---

8. Conclusion

In this project, we compared three classification methods, Super Learner, Optimized Super Learner and Deep Learning, for detecting plastic explosives in suitcases. All methods performed well, but we observed the following:

Machine Learning – SuperLearner			
Bomb screening		**Reality**	
		Explosives	No explosives
Predicted	Explosives	382 (TP)	0 (FP)
	No explosives	0 (FN)	216 (TN)

Deep Learning – Neural networks			
Bomb screening		**Reality**	
		Explosives	No explosives
Predicted	Explosives	374 (TP)	3 (FP)
	No explosives	0 (FN)	221 (TN)

- **Super Learner** outperformed the deep learning model in terms of accuracy and misclassification rate. Its performance on the test set showed a lower misclassification rate 0% compared to deep learning 0.5% on the validation. The training accuracy results for CV Super Learner and Genetic Algorithm optimized Super Learner are very similar. However, since CV Super Learner predicts 1 False negative we choose the GA optimized Super Learner as our final selection.

- **Deep Learning** showed strong performance as well, but the model's interpretability is limited compared to the Super Learner. The neural network approach may be harder to interpret, making it less suitable for applications where explainability is important.

Given these results, we recommend using the **Super Learner(GA optimized)** for this problem, especially in scenarios where model interpretability is crucial, although deep learning could still be considered for its ability to capture complex patterns.

---

9. Future Work

- **Hyperparameter Tuning**: Further hyperparameter tuning of both the Super Learners and also for deep learning models could improve performance. Hyperparameter tuning using genetic algorithms is also an exciting idea to look into.
- **Feature Engineering**: Additional feature engineering could improve model performance, particularly for deep learning models, which often benefit from richer feature representations.
- **Model Interpretability**: Future work could focus on improving the interpretability of the deep learning model, for example by using techniques like SHAP (Shapley Additive Explanations).

---

10. References

- Cabrera, S., & McDougall, R. (2002). Detection of plastic explosives in suitcases using X-ray signals. *Journal of Applied Statistics*.
- R Documentation: SuperLearner package, Keras in R, and caret package.
- Chris Kennedy.(2017), Guide to SuperLearner, *University of California, Berkeley*

---

11. Appendix

R Code and all datasets used/created are in the zipped folder provided.